Logger

Titel:
Title:

A client/server based logging system

Dokumenten Typ:
Document Type: User Manual

Konfigurations-Nr.:
Configuration Item No.:

Referenz- Nr.:
Reference No.:

Klassifikations-Nr.:
Classification No.:

Lieferbedingungs-Nr.:
DRL/DRD No.:

Freigabe Nr.:
Release No.:

Gruppierung (Dok.):
Group (Doc.-related):

Gruppierung (Version):
Group (Version-related):

Thema:
Subject:

Kurzbeschreibung:
Abstract:

The document describes a client/server based logging system that was developed for the CGS project, but may be used in other software systems as well.

Autor: Prepared by:	CGS Team	**Org. Einh.:** Organ. Unit:	TE 55	**Unternehmen:** Company:	Astrium Bremen
Geprüft: Agreed by:		**Org. Einh.:** Organ. Unit:		**Unternehmen:** Company:	
Genehmigt: Approved by:		**Org. Einh.:** Organ. Unit:		**Unternehmen:** Company:	
Genehmigt: Approved by:		**Org. Einh.:** Organ. Unit:		**Unternehmen:** Company:	

Copyright by EADS - All Rights Reserved

Daten/Dokument-Änderungsnachweis/Data/Document Change Record (DCR)

Ausgabe Issue	**Datum** Date	**Betroffener Abschnitt/Paragraph/Seite** Affected Section/Paragraph/Page	**Änderungsgrund/Kurze Änderungsbeschreibung** Reason for Change/Brief Description of Change
1/-	2007-08-01		Initial version

Copyright by EADS - All Rights Reserved

Inhaltsverzeichnis/Table of Contents (ToC)

1 Reference Documents 5
1.1 An XML Based Configuration Concept, Configuration Editor and APIs. CGS-RIBRE-MA-0008, 1/-, 2006-10-01 5

2 Overview 6

3 Usage of the Logging System 7
3.1 The Message Server 7
3.2 The Message Window 8
3.2.1 Command Line Invocation 8
3.2.2 Window Description 10
3.2.3 The Message Details Window 12
3.2.4 The Menu Bar 13
3.2.5 The Tool Bar 15
3.2.6 Filters 16
3.2.7 Views 18
3.2.8 Styles 19
3.2.9 Acknowledging Messages 20
3.2.10 Searching in Messages 20
3.2.11 Preferences 20
3.2.12 Sending Messages 21

4 Configuration of the Message Window 22
4.1 Configuration Concept 22
4.2 Overall Configuration Structure 23
4.3 Basic Message Window Configuration Layout 23
4.4 The `window` Group 24
4.5 The `views` Group 25
4.6 View Definitions 25
4.7 Column Definitions 26
4.8 The `filters` Group 27
4.9 Filter Definitions 27

5 Internals 28
5.1 The Message File Format 28
5.2 The Logger Protocol 29
5.2.1 General Format 29
5.2.2 Connection/Disconnection 29
5.2.3 Status Notification 29
5.2.4 Issuing Messages 30

Copyright by EADS - All Rights Reserved

5.2.5 Acknowledging Messages 30
5.2.6 Message Subscription 30
5.2.7 Message Delivery 30
5.2.8 Summary Request 31
5.2.9 Client/Client Communication 31
5.2.10 Session Notification 31

Copyright by EADS - All Rights Reserved

1 Reference Documents

1.1 An XML Based Configuration Concept, Configuration Editor and APIs.
CGS-RIBRE-MA-0008, 1/-, 2006-10-01

Copyright by EADS - All Rights Reserved

2 Overview

The logging system has a client/server based architecture: A central Message Server collects messages from senders and distributes them to reader clients (typically message windows). All messages are stored in files. Message clients can both send and receive messages. They may run on different computers in a network and communicate with the Server through APIs. Currently there are APIs for Ada and Java.

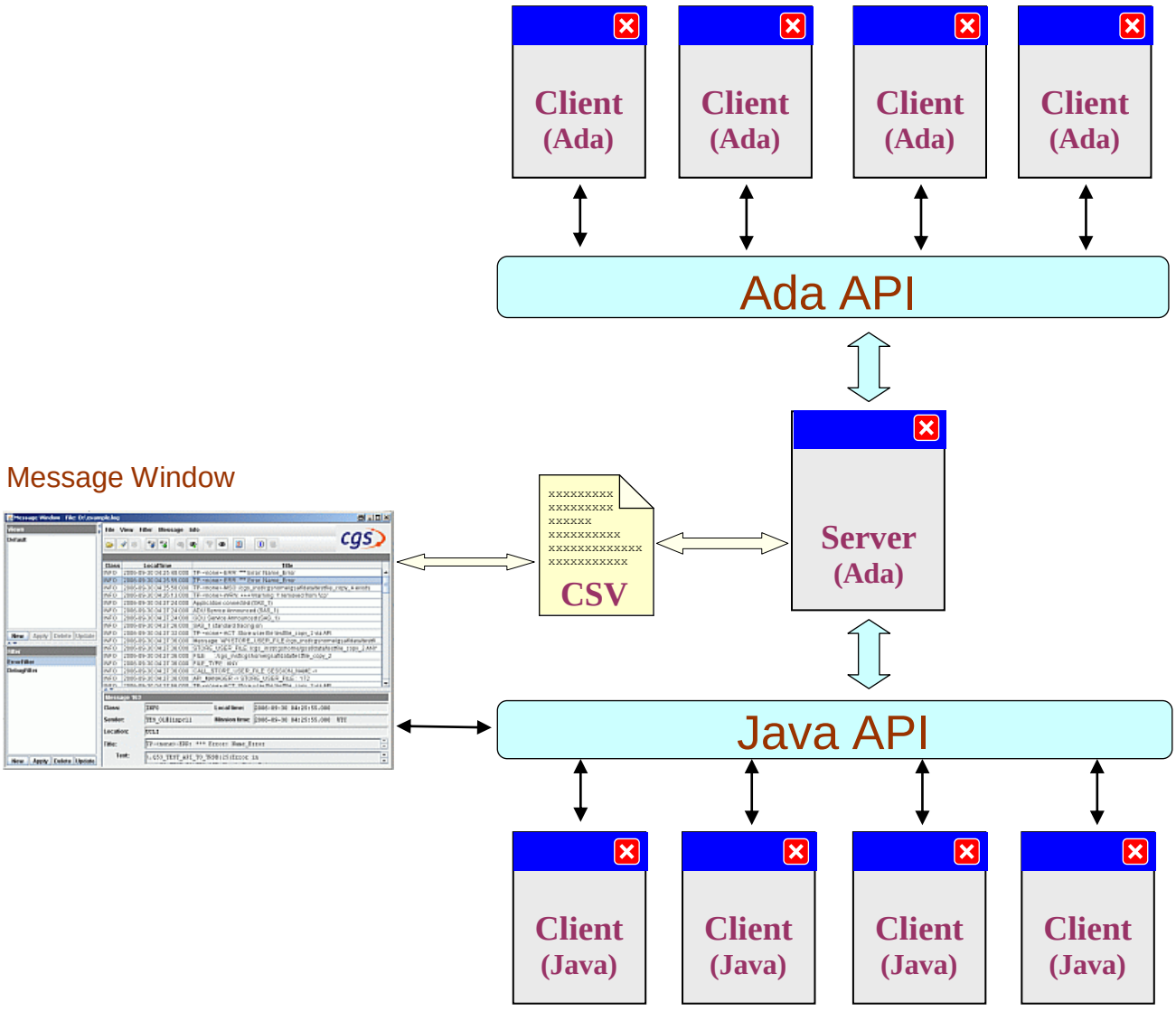
Within a software system there may be more than one message server. A client can, however, connect to only one server at a time.

Message Window

The Message Window is a Java client. It connects to a server and displays all messages received from the server. But Message Window clients can also directly read message files and display their contents.

The Message Window allows to display the messages in different formats, sort messages according to different criteria, search for message contents etc. Different views and filters may be defined, given a name and stored for later reuse.

The Message Window can be configured through XML based configuration files.



Copyright by EADS - All Rights Reserved

3 Usage of the Logging System

3.1 The Message Server

...

Copyright by EADS - All Rights Reserved

3.2 The Message Window

3.2.1 Command Line Invocation

The Message Window is started from the command line with the command

`message_window [`*`config_file...`*`] [options]`

`config_file...`

the names of the XML configuration files to be used to configure the Message Window.
Several files can be given, they are accumulated, as described in the document in section 1.1. The definitions of these files cannot be changed! In addition to the files given, a user configuration file can be defined with the `user_config` option. The definitions of this file can be changed!

Options:

`-help ["…"]`

Print help text and exit message window afterwards. When the optional parameter "…" is set, the help is printed more verbosely.

`-user_config` *`file_name`*

This option sets the user's configuration file which will be loaded. The definitions of this file can be changed.

A user configuration file can be also loaded interactively (Settings -> Open Configuration…). So it is not mandatory to load it per command line option.

`-server` *`host_name:port`*

Defines the host name and port number of the message server.
If omitted, `localhost:7250` is assumed.

`-connect`

Connect immediately to the server. This option can't be used in conjunction with the file option.

`-file` *`filename`*

Open the message log file specified by filename. This option can't be used in conjunction with the connect option.

`-incoming_filter [`*`filtername | filter_definition`*`]`

This option defines a filter to load only those messages, which match the filter, into the Message Window. An already defined filter name (from config files) can be used or a textual filter description defines a temporary filter which name is displayed as '#in_temp#'. An incoming filter cannot be changed or disabled after start.

A textual filter description is composed of three optional components separated by commas. The components are AND associated:

`class_filter, string_patterns, time_filter`

`class_filter` is an enumeration of all matching message classes, e. g.:

`class: (INFO, WARNING)`

`string_patterns` are regular expressions with not (!), and (&), or (|) and case-insensitive indicators, e. g.:

`!title:/`*`expr`*`/i & host:/`*`expr`*`/ | text:/`*`expr`*`/`

`time_filter` starts with an LT or MT tag followed by a time range.:

`LT:` *`first .. last`*

`If first or last is omitted, begin or end of time range is assumed.`

Copyright by EADS - All Rights Reserved

Examples:

```
class:(WARNING,ERROR)
!title:/ucl/i & text:/compiler/
class:(WARNING,ERROR), !title:/ucl/i & text:/compiler/
class:(WARNING,ERROR), LT:2007-01-01 00:00 .. 2007-01-01 12:00:00
```

`-display_filter [`*`filtername | filter_definition`*`]`

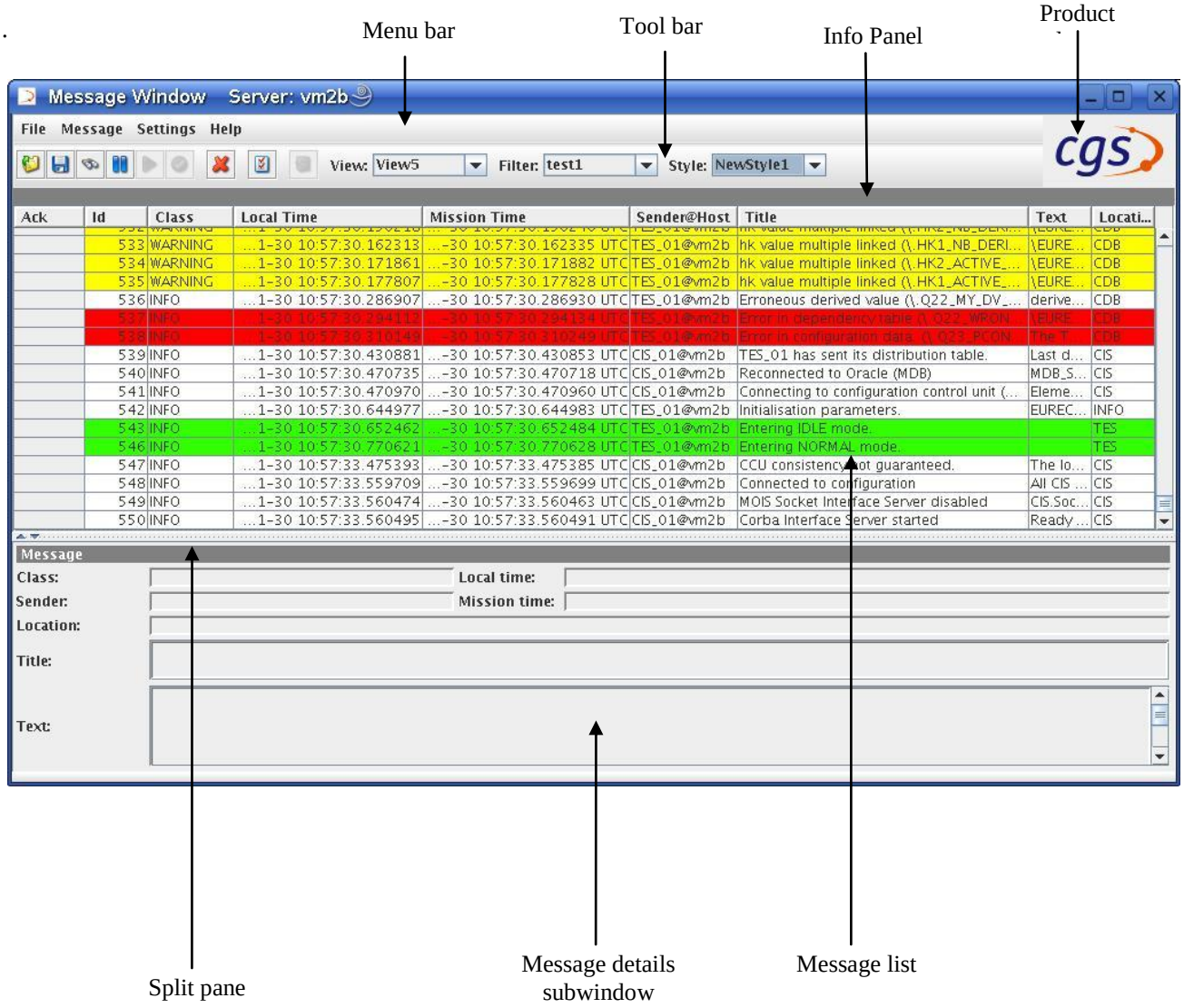
This filter works on the already loaded messages. Only those messages, which match the filter, are shown. A display filter can be changed after start.

For the option values the description of "-incoming_filter".

Copyright by EADS - All Rights Reserved

3.2.2 Window Description

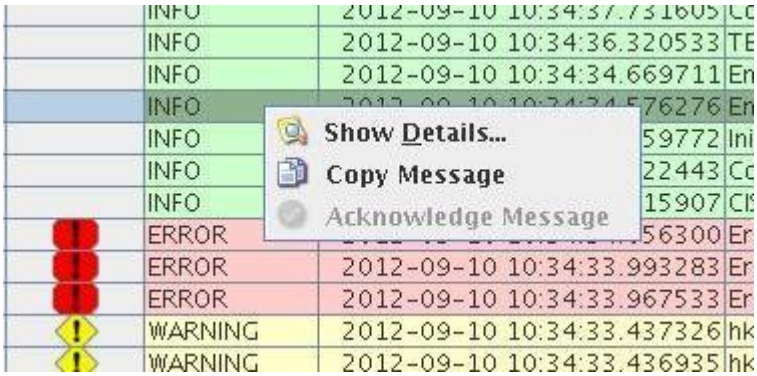
The picture shows the layout of the window used by the Message Window program.



The Message Window consists of a menu bar, a tool bar, info panel, and a content area. The content area is separated by a split pane and shows the messages in a list and the selected message on the message details pane. By clicking on the tiny arrows on the divider the message list or message details pane can be hidden or expanded. Drag the dimpled line to change the space used for the message details pane. The info panel shows the applied filter and view.

The window title indicates the current message source. The sources can be server, group (multicast), or file.

Copyright by EADS - All Rights Reserved



Clicking the right mouse button pops up a menu from which you can choose some options.

A new one is to copy the contents of the message into the clipboard by pressing the *Copy Message* menu item.

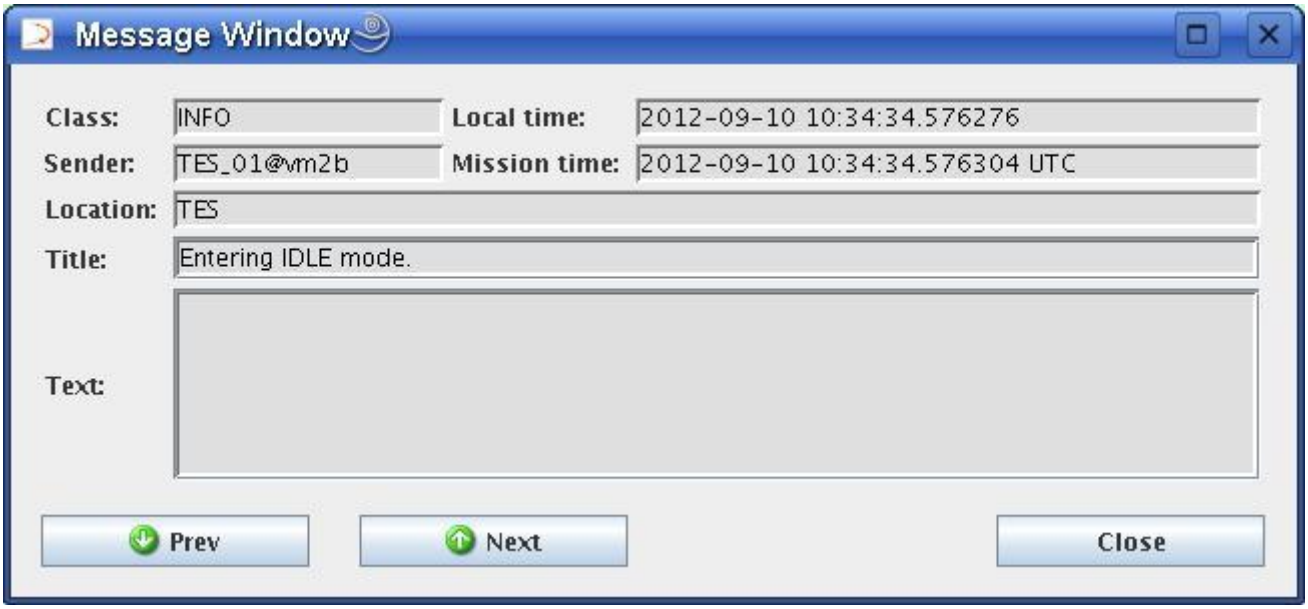
The message text in the clipboard has the following format:

<class>, <local time>, <mission time>, <sender>, <location>, <title>, <text>

Copyright by EADS - All Rights Reserved

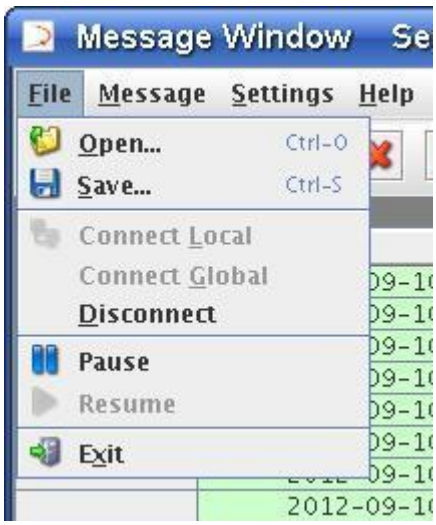
3.2.3 The Message Details Window

Double-clicking a message in the message list opens a separate details window. It contains the same information as the message details pane, but offers the possibility to navigate to the next/previous message by clicking the adequate button.



Copyright by EADS - All Rights Reserved

3.2.4 The Menu Bar



The *File* Menu

Open	Open a message file and display the messages contained in it. A connection to a message server is closed before loading the file. With clicking the check button you can load "hidden" messages (e.g. HLCL commands).
Save	Save the currently displayed messages in a file.
Connect local	Connect to the message server.
Connect global	Join the CGS group (multicast) to listen for messages.
Disconnect	Disconnect from message server.
Pause	Do not show incoming messages
Resume	Show all messages again (including the incoming messages during the Pause).
Exit	Exit the program

The *Message* Menu

Show details	Show the details of the selected message from the message list.
Find	Search messages, opens a search form.
Send	Send a message to the message server.
Acknowledge	Acknowledge message.
Copy	Copy message to clipboard.
Clear	Clear the message list.

Copyright by EADS - All Rights Reserved

The *Settings* Menu

Views	Open the views dialog.
Filters	Open the filter dialog.
Styles	Open the styles dialog.
Preferences	Open the preferences dialog
Open Configuration	Open file dialog for opening user configuration file.
Save Configuration	Open file dialog for saving user configuration file.
Exit Dialog	Shows Exit Dialog (or not) on quit.

The *Info* Menu

About	Show information about the program.
Open User Manual	Open the manual in pdf format with acroread.

Copyright by EADS - All Rights Reserved

3.2.5 The Tool Bar

The tool bar contains a number of symbols that denote frequently used functions and, thus, allows easier access to these functions. Most of them are also available in the menu bar.

Open File: Open an existing message file. A file chooser will be displayed to select the file.

Save File: Save the currently displayed messages in a file.

Search Messages: Search for patterns (given as regular expressions) within the currently maintained messages.

Pause: Pause updating the message list, don't show new messages (freeze mode).

Resume: Resume updating the message list (stop freeze mode).

Acknowledge Message: Acknowledge the selected message

Clear View: Clear the messages list.

Preferences: Open the preferences dialog

Server summary: Display a server summary (not implemented yet)

View *View: Select a view definition*

Filter *Filter: Select a filter definition.*

Style *Style: Select a style definition.*

Copyright by EADS - All Rights Reserved

3.2.6 Filters

Filters can be used to reduce the amount of messages displayed. Several filters can be defined, given a name and kept in a filter list, from where they can be selected and applied.

The configuration file(s) loaded by the program can predefine named filters; they are shown in the filter list. Only those filters that were created by the user or loaded from the user's own configuration file can be modified. Filters from other configuration files are protected against modifications.

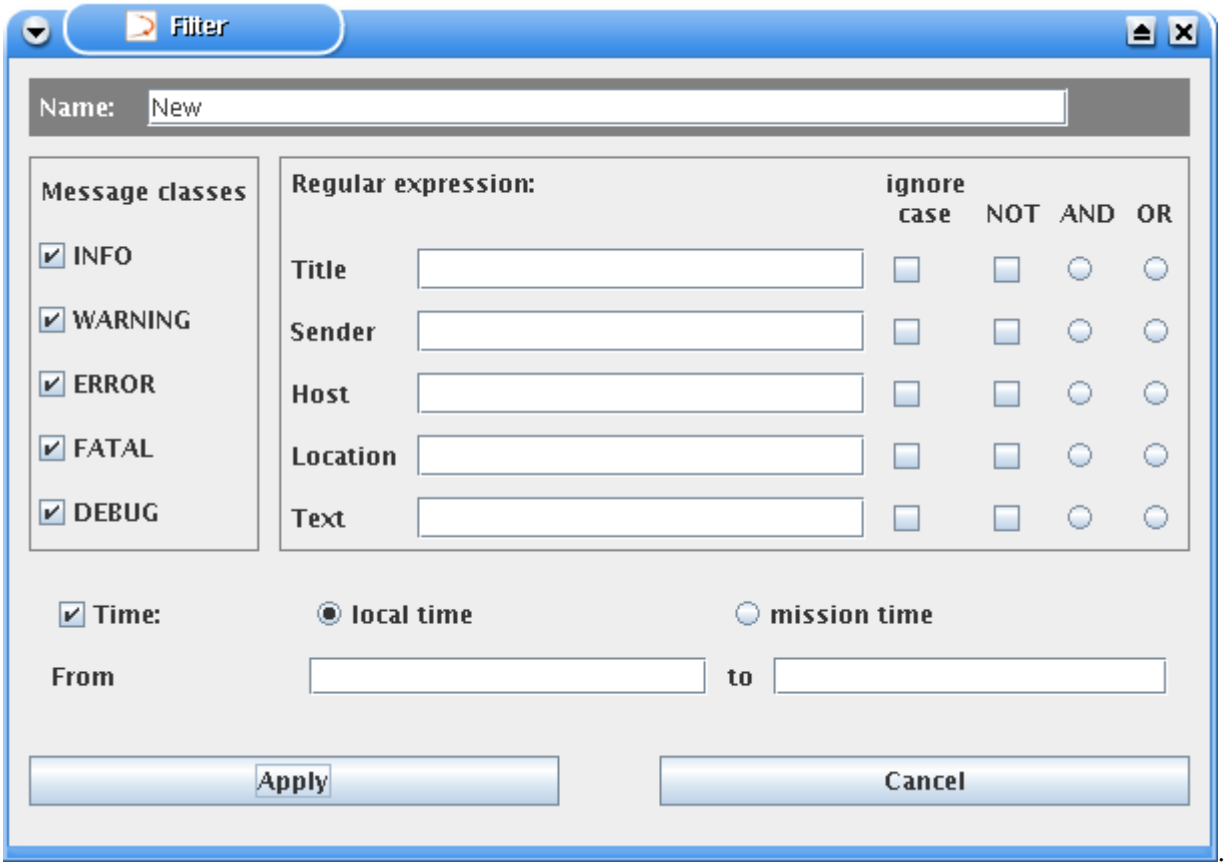
The *current filter* (i.e. the one currently applied) is stored as part of the user configuration on program termination. It is automatically re-applied when starting the program again.

To modify filters, the filter definition form is used. It is shown in the picture.

The symbols before the filter definition name show
a user filter definition (here: my_filter) is editable, the other definitions are locked.
Locked definitions cannot be edited, but they can be copied (right mouse click).

The same symbols are used in the windows: "Manage Views" and "Manage Styles".

Copyright by EADS - All Rights Reserved



A filter definition consists of three sections. These sections are AND associated:

- Message classes: This is a subset of the message classes that are to be displayed.
- Regular expressions for the message text fields: These can be case sensitive or insensitive. If the NOT flag is selected, the pattern matches if it does not appear in the text field, otherwise it matches if the pattern is present. Single patterns can be AND or OR associated. The logical expression is formed by first giving the AND associated patterns and then the OR associated ones.
- Time ranges for local time and mission time: A time specification has the form `YYYY-MM-DD hh:mm:ss`. A time range may consist of a lower or upper bound only, i.e. one of the bounds may be omitted.

The *Apply* button applies the filter to the filter list.

Copyright by EADS - All Rights Reserved

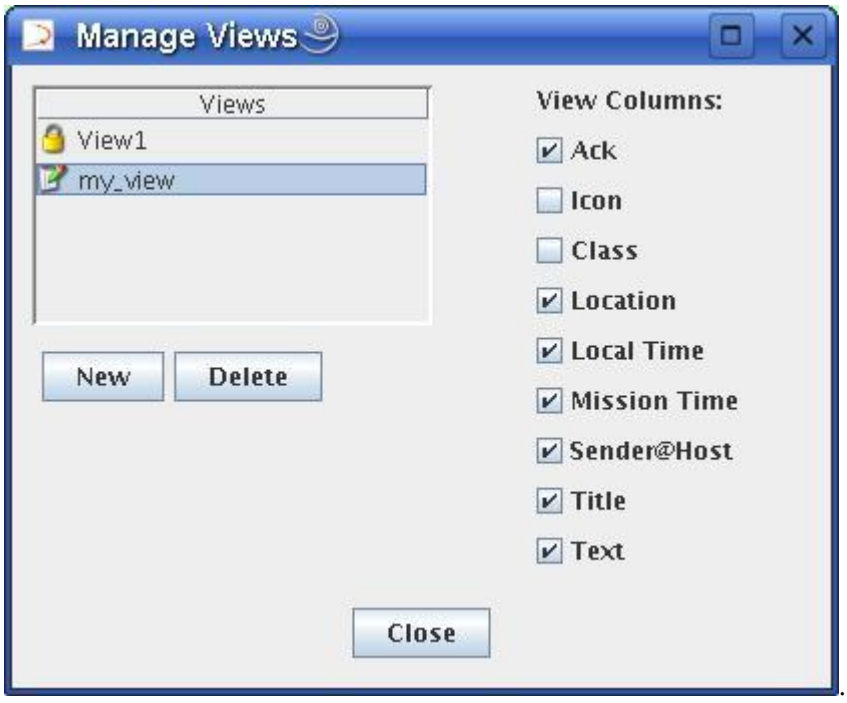
3.2.7 Views

A *view* defines the appearance of the message list. It comprises the subset of columns displayed, the position and width of the columns, and the ordering of the messages in the list.

Messages can be sorted by several columns. A double click on a column header selects the column as the primary sorting criterion, a single click on one or more other columns adds them as subordinate sorting criteria. The first click sets ascending order, subsequent clicks toggle between ascending and descending order for the particular column. The sorting order is indicated with an arrow in the column header, a bigger arrow for the primary column, and smaller arrows for subordinate columns. The messages in the picture are sorted first by Sender@Host, then by Local Time, both in ascending order.

	ID	Class	...	LocalTime	...	Sender@Host	Title
	13	INFO	...	2006-09-28 ...		CGS@linpc11	
	17	INFO	...	2006-09-30 ...		DBS_01@linpc11	Session closed.
	1	INFO	...	2006-09-30 ...		SAS_1@linpc11	SAS_1 standard tracing on
	2	INFO	...	2006-09-30 ...		SAS_1@linpc11	Message: 'API STORE_USER_FILE /cgs_inst/cgsho...
	3	INFO	...	2006-09-30 ...		SAS_1@linpc11	STORE_USER_FILE: /cgs_inst/cgshome/gsaf/data...

The position of the columns can be changed by dragging the column header with the left mouse button to the desired position. The width of the columns can be changed by pulling the right border of the column header with the left mouse button to the desired width. The subset of columns to be displayed by using the view definition form:



Views can be given a name and kept in a view, from where they can be selected and applied. The configuration file(s) loaded by the program can predefine named views; they are shown in the view list. The user defined views are saved in the user configuration file on program termination. Only those views that were created by the user or loaded from the user's own configuration file can be modified. Views from other configuration files are protected against modifications.

The *current view* (i.e. the one currently applied) is stored as part of the user configuration on program termination. It is automatically re-applied when starting the program again.

Copyright by EADS - All Rights Reserved

3.2.8 Styles

A style defines the appearance of a set of messages. They can have special attributes like colour, a symbol or a jingle.

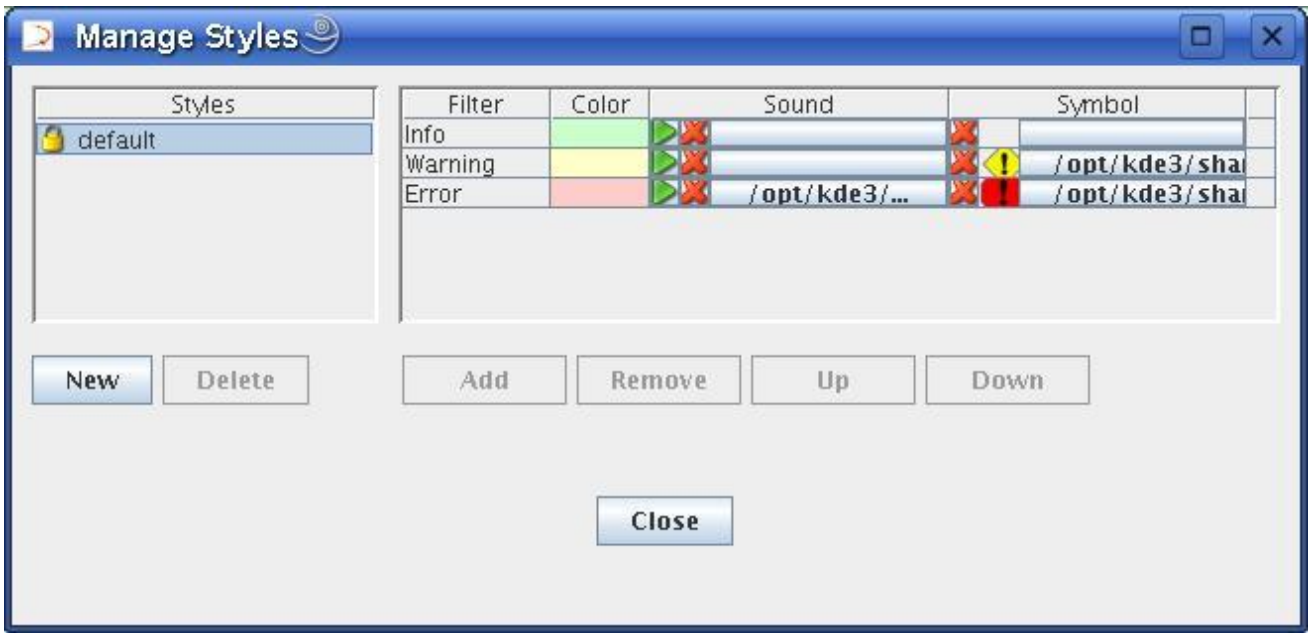
Opening the style window new style can be defined or old styles can be changed or deleted.

If an existing style is selected at the left table, on right table the corresponding attribute definitions are displayed.

You can define a color, a symbol and a sound for a certain kind of message. The kind of message is defined by a filter (first column).

A style definition can have multiple attribute definitions. These are ordered from top to bottom. For example: if a message is from the first kind (filter: Hallo) it is red displayed, even it is also from another kind later in the list.

The order of the attributes can be changed by the Up- and Down-Buttons.



Copyright by EADS - All Rights Reserved

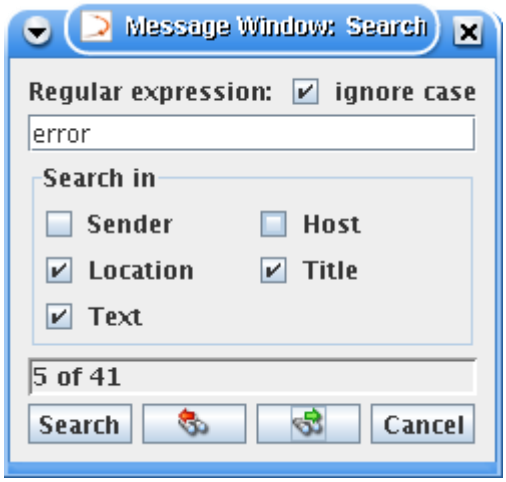
3.2.9 Acknowledging Messages

Messages may require acknowledgement by the user. Such messages have an acknowledgement status indicator in their leftmost column: a red exclamation mark, if the message is to be acknowledged, a small checked box if it has already been acknowledged:

!	1	FATAL	...	2007-07-17 08:3...		CGS@vm3a-227	Give me an acknowledge!
☑	1	FATAL	...	2007-07-17 08:3...		CGS@vm3a-227	Is acknowledged

A message can be acknowledged by selecting the message and clicking on the acknowledge button in the tool bar. This will send an acknowledgement message to the message server, which keeps track of all acknowledgements.

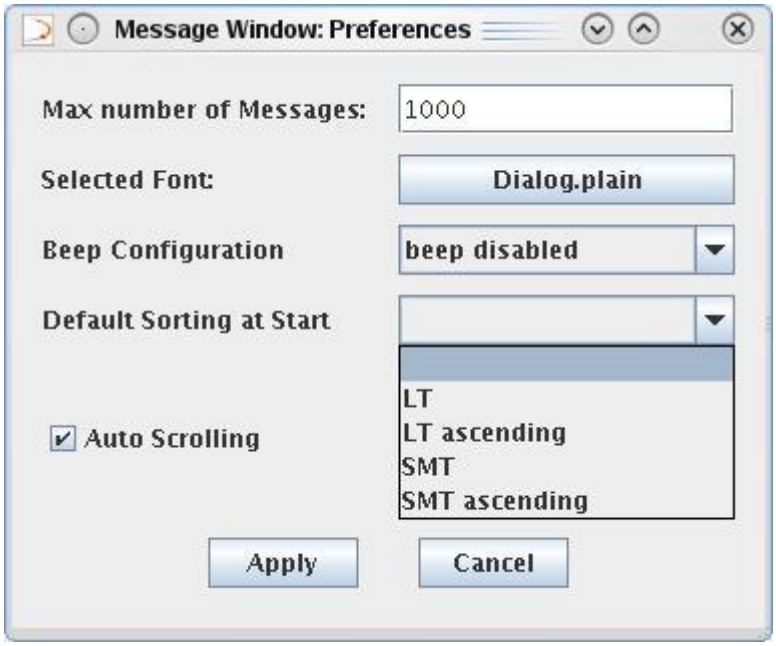
3.2.10 Searching in Messages



Messages can be searched for certain contents in their text fields Clicking on the search button in the tool bar or selecting the Search item from the Message menu will open the search dialog.

A regular expression can be applied to a selected set of text fields. Clicking on the *Search* button starts the search. The number of occurrences is displayed, and the search arrows can be used to move backward and forward in the list of messages found, this selects the corresponding message in the list and highlights it.

3.2.11 Preferences

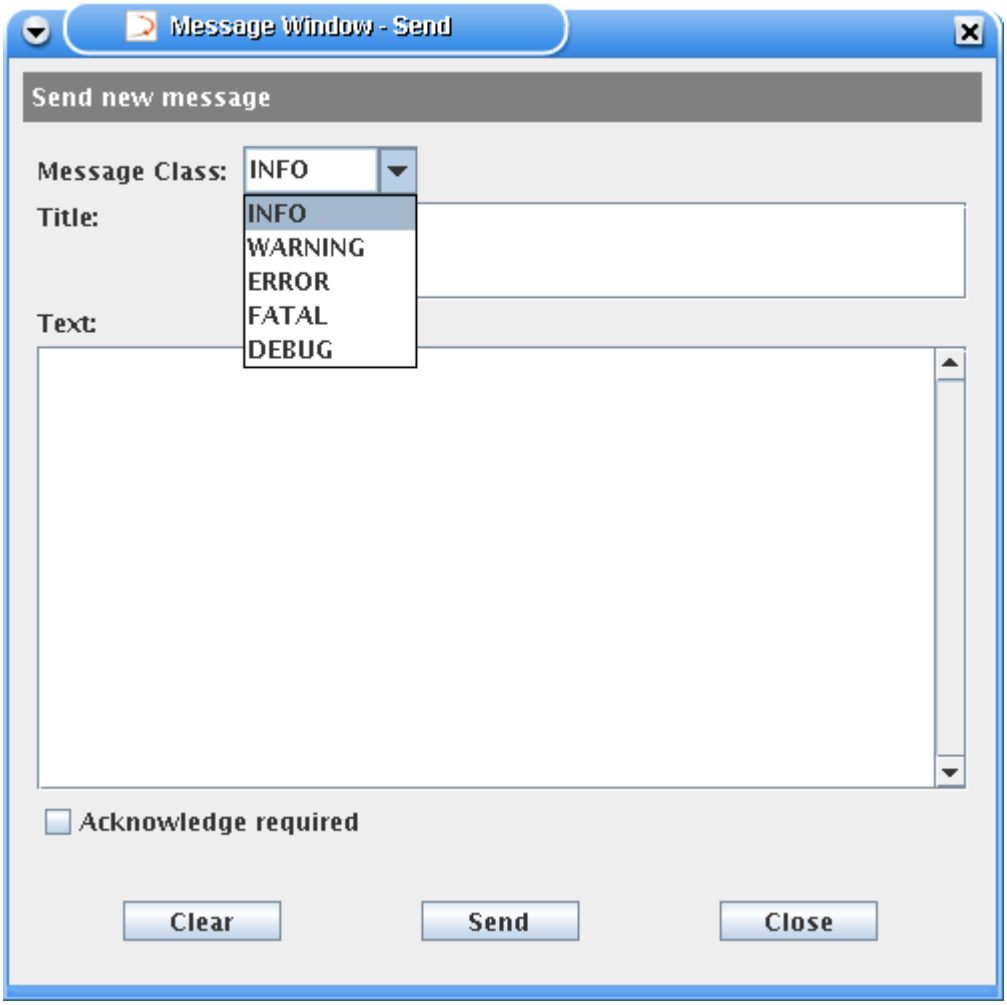


1. The maximum number of messages
2. Selected Font
3. "Simple" beep configuration (relevant for systems without sound card or speaker)
4. Default sorting at start
5. Whether auto scrolling is enabled.

Copyright by EADS - All Rights Reserved

3.2.12 Sending Messages

The user of the Message Window can send messages to the message server by selecting the *Send* item from the *Message* menu.



Copyright by EADS - All Rights Reserved

4 Configuration of the Message Window

4.1 Configuration Concept

The Message Window is configured through one or more XML files; a configuration comprises the following elements:

- a version number
- the window appearance and preferences:
  - width and height of the window
  - position of the window
  - the maximum number of messages kept and displayed
  - whether auto scrolling is enabled
  - whether exit needs a confirmation
- a list of view definitions
- a list of filter definitions

The configuration is based on the configuration concept described in 1.1. The Configuration Editor can be used to set up and edit the configuration.

Please note that a configuration need not be given in one file, but may be split in several files, each containing a part of the overall configuration. Only when merged together, the set of separate files must result in a complete Message Window configuration. All files must be given as parameters to the `message_window` program. If a file given in the command line contains attributes that were already given in a preceding file, these multiple attributes are resolved as follows:

- View definitions from different files are gathered in the list of views. View names must be unique. If the same view name is given twice, an error message is displayed.
- Filter definitions from different files are gathered in the list of filters. Filter names must be unique. If the same filter name is given twice, an error message is displayed.
- Window related attributes overlay and hide the previously given attributes.

This allows to build multiple-level configurations, e. g. in the form that a basic configuration on system level may be extended and modified by separate configurations on group and user level. This way, a user may not only add his own views and filters, but also modify e.g. the window size.

The values of several attributes may contain placeholders for environment variables. They have one of the two forms

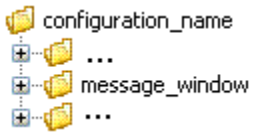
```
{$variable_name}
{$variable_name=default_value}
```

These placeholders are substituted by the Message Window with the value of the corresponding environment variable. If the variable does not exist and a default value is given, the default value replaces the placeholder. If the variable does not exist and there is no default value, the Message Window will report an error and reject the configuration.

The following describes the configuration, giving both a graphical representation of the elements in Configuration Editor conventions (in terms of groups and attributes) and a pure XML pattern. This should enable a reader to manually set up a configuration in XML form. Within the XML patterns, identifying attributes to be given in this form are shown in bold, attribute values to be supplied by the user are shown in italic and in red text.

Copyright by EADS - All Rights Reserved

4.2 Overall Configuration Structure

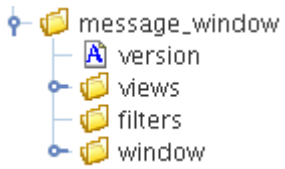



```
<?xml version="1.0" encoding="UTF-8"?>

<configuration name="configuration_name">

  ...
  ...  possibly other configuration elements
  ...

  <group name="message_window">
    ...
    ...  all Message Window configuration goes here
    ...
  </group>

  ...
  ...  possibly other configuration elements
  ...

</configuration>
```

The overall configuration must contain a global group named `message_window`, this group must contain the Message Window configuration. Any configuration elements outside this group are ignored. This allows to keep a Message Window configuration as part of a general configuration.

The `<configuration>` element may or may not reference a Schema file, In any case, each XML file will be validated, both by the Configuration Editor and by the APIs (and hence by the Message Window), against the basic Schema on which the configuration concept is based.

4.3 Basic Message Window Configuration Layout



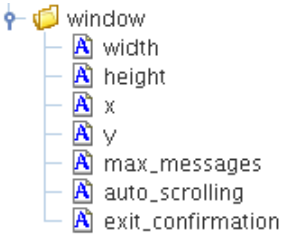

```
<group name="message_window">
  <attribute name="version">
    <value>1.0.0</value>
  </attribute>
  <group name="window"> ... </group>
  <group name="views"> ... </group>
  <group name="filters"> ... </group>
</group>
```

The `message_window` group must contain a version attribute and three subgroups:

`version`	Identifies the version of the configuration file structure
`window`	contains all attributes that determine the appearance of the Message Window window
`views`	contains a list of view definitions
`filters`	contains a list of filter definitions

Copyright by EADS - All Rights Reserved

4.4 The `window` Group



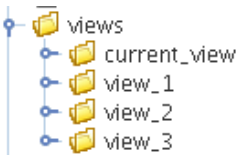

```
<group name="window">
  <attribute name="width"><value>integer</value></attribute>
  <attribute name="height"><value>integer</value></attribute>
  <attribute name="x"><value>integer</value></attribute>
  <attribute name="y"><value>integer</value></attribute>
  <attribute name="max_messages"><value>integer</value></attribute>
  <attribute name="auto_scrolling"><value>boolean</value></attribute>
  <attribute name="exit_confirmation"><value>boolean</value></attribute>
</group>
```

The `window` group comprises a number of attributes that determine the appearance of the Message Window window:

`width`	an integer defining the width of the window (in pixels) if omitted, a default width is assumed
`height`	an integer defining the height of the window (in pixels) if omitted, a default height is assumed
`x`	Horizontal position of the window.
`y`	Vertical position of the window.
`max_messages`	an integer defining the maximum number of messages kept by the Message Window if omitted, the number of messages kept is not limited (may cause memory problems)
`auto_scrolling`	Indicates whether auto scrolling of messages is enabled.
`exit_confirmation`	Indicates whether an exit confirmation is needed.

Copyright by EADS - All Rights Reserved

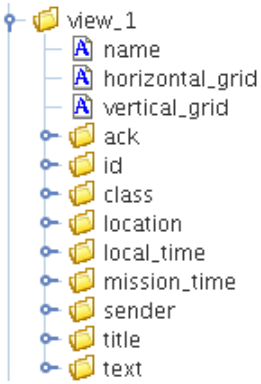
4.5 The `views` Group




```
<group name="views">
  <group name="current_view">...</group>
  <group name="view_1">...</group>
  <group name="view_2">...</group>
  <group name="view_3">...</group>
  ...
</group>
```

The `views` group contains a list of view definitions. Each view definition is described by a subgroup. The `current_view` group contains the currently applied view. This view is automatically applied when opening the Message Window. All view definition groups have the same structure, as described in the following subsection:

4.6 View Definitions




```
<group name="view_N">
  <attribute name="name"><value>name<value></attribute>
  <group name="ack">...</group>
  <group name="id">...</group>
  <group name="class">...</group>
  <group name="location">...</group>
  <group name="local_time">...</group>
  <group name="mission_time">...</group>
  <group name="sender">...</group>
  <group name="title">...</group>
  <group name="text">...</group>
</group>
```

A view definition defines the columns of the table displaying the messages.

`name`	the name of the view displayed in the list of views
`ack`	the acknowledgement column
`id`	the message ID column
`class`	the message class column
`location`	the location column
`local_time`	the local time column
`mission_time`	the mission time column
`sender`	the sender column
`title`	the title column
`text`	the text column

Copyright by EADS - All Rights Reserved

4.7 Column Definitions

```
<group name="column_name">
  <attribute name="name">
    <value>name<value>
  </attribute>
  <attribute name="on">
    <value>boolean<value>
  </attribute>
  <attribute name="width">
    <value>integer<value>
  </attribute>
  <attribute name="position">
    <value>integer<value>
  </attribute>
</group>
```

A column definition defines the appearance of a column in the message table.

`name`	the name of the column displayed in the table header
`on`	true: the column is shown in the message table false: the column is hidden
`width`	width of the column in pixels
`position`	position of the column in the message table, counted from left to right, starting at 1

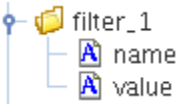
Copyright by EADS - All Rights Reserved

4.8 The `filters` Group

```
<group name="filters">
  <group name="current_filter">...</group>
  <group name="filter_1">...</group>
  <group name="filter_2">...</group>
  <group name="filter_3">...</group>
  ...
</group>
```

The `filters` group contains a list of filter definitions. Each filter definition is described by a subgroup. The `current_filter` group contains the currently applied filter. This filter is automatically applied when opening the Message Window. All filter definition groups have the same structure, as described in the following subsection:

4.9 Filter Definitions




```
<group name="filter_1">
  <attribute name="name">
    <value>Filter One</value>
  </attribute>
  <attribute name="value">
    <value>class:(ERROR,WARNING,FATAL,DEBUG),|host:/linsrv-1/i</value>
  </attribute>
</group>
```

Definition of a filter:

`name`	the filter name shown in the filter list
`value`	filter definition as described in section 3.2.1 Command Line Invocation

Copyright by EADS - All Rights Reserved

5 Internals

5.1 The Message File Format

Message files are encoded in CSV format, each message **as one line** with each message component as one column. The first line of each message file is a CSV encoded list of column names. This can be used by the logger software to perform version and consistency checks.

The following fields are encoded in this order:

`CLASS`	the message class (see below)
`LOCAL_TIME`	the local time in the form `YYYY-MM-DD hh:mm:ss.nnnnnnnnn` `nnnnnnnnn` = parts of seconds with 9 digits (nano seconds accuracy)
`MISSION_TIME`	the mission time in the form `YYYY-MM-DD hh:mm:ss.nnnnnnnnn SSS` `nnnnnnnnn` = parts of seconds with 9 digits (nano seconds accuracy), `SSS` = time scope (`UTC`, `GPS`)
`SENDER`	for normal messages: the sender application in the form `application@host` for acknowledgement messages: the user information in the form `user@host`
`LOCATION`	an additional sender information, e. g. a location within the sender program
`ACKNOWLEDGE`	`TRUE` or `FALSE` indicating whether the message is to be acknowledged
`TITLE`	the title of the message
`TEXT`	the message text
`DISPLAY`	TRUE or FALSE indicating whether the message shall be displayed

Characters of TEXT column are escaped using Java string rules to avoid line feeds in columns. All other columns must not contain CR or LF characters.

Message Classes

Messages are classified with the following classes:

`INFO`	an informal message
`WARNING`	a warning message
`ERROR`	a non-fatal error message
`FATAL`	a fatal error message
`DEBUG`	a debug message
`ACK`	a message acknowledgement, refers to another existing message

Copyright by EADS - All Rights Reserved

5.2 The Logger Protocol

5.2.1 General Format

Server and clients communicate via textual messages. The general format is

```
<TAG>.<SUBTAG> <parameter> ...
```

Each message consists of a *tag* and possibly a *subtag* that indicate the kind of message. They are followed by parameters that are specific to the message kind.

5.2.2 Connection/Disconnection

In order to connect to a logger server, clients send a `CONNECT/INTERCONNECT` request to the server. The server replies with either a positive acknowledgement (`CONNECTED`, `DISCONNECTED`) or an error message (`ERROR`): A client connected as INTERCONNECT client will receive status notifications and messages from other clients. It can subscribe for messages and receive reports. A client connected via CONNECT can send messages only.

Connection as Client (client → server)

```
CONNECT <client name>
```

```
INTERCONNECT <client name>
```

Disconnection (client → server)

```
DISCONNECT
```

Success Response (server → client)

```
CONNECTED <client id>  <message structure>
```

```
DISCONNECTED
```

The `<message structure>` is a CSV encoded list of message column headers (names). It defines the message structure understood by the server. All messages sent by senders and delivered to receivers contain values in the same structure. This allows a consistency check between server and clients.

Error Response (server → client)

```
ERROR <error text>
```

5.2.3 Status Notification

Whenever there is a status change, the server notifies all connected clients with a status notification.

Notification of Client Connections and Disconnections (server → client)

`CLIENTS <client name>, <host>, <id>  ...` (comma separated list of connected client information triples)

Notification of Error (server → client)

```
ERROR <error text>
```


Copyright by EADS - All Rights Reserved

5.2.4 Issuing Messages

Server clients can send messages with a `SEND` request.

Send Message (client → server)

```
SEND <message>
```

The `<message>` is a CSV encoded list of message components. The structure is identical to the `<message structure>` returned with the connection acknowledgement. Since the server does automatically add the message id, the field must be really empty; not 0, "0", or "". The id field is always required (even if empty) and must be the first one in the character separated value.

Justification: This has been defined for performance reasons, since the server can directly deliver the message id concatenated with the received message to its subscribers without parsing it.

5.2.5 Acknowledging Messages

Server clients can acknowledge messages with an `ACKNOWLEDGE` request.

Acknowledge Message (client → server)

```
ACKNOWLEDGE <message>
```

The `<message>` is a CSV encoded list of message components. The structure is identical to the `<message structure>` returned with the connection acknowledgement with the following specifics:

- The ID is the ID of another existing message.
- The class is `ACK`.
- The sender is an identification in the form *user@host*.
- Local time and mission time set accordingly.
- Other fields are empty.

5.2.6 Message Subscription

Connected receiver clients may subscribe to message delivery.

Subscription (client → server)

```
SUBSCRIBE [<filter string>]
```

Subscription Acknowledgement (server → client)

```
SUBSCRIBED [LCM_ID:<message id>]
```

Unsubscription (client → server)

```
UNSUBSCRIBE
```

The `<filter string>` is a CSV encoded list of filter components (message classes, time span, regular expressions for message parts). Details TBD

The SUBSCRIBED message can contain an optional identifier for the message id of the last message in the server cache (LCM_ID = last cached message identification) followed by column character and the message id.

5.2.7 Message Delivery

When a receiver client has subscribed to message delivery, the server will send first all messages from internal server cache and then the server will continuously deliver messages, until the end of messages has been reached (client unsubscribed or all messages from requested time span have been delivered).

Copyright by EADS - All Rights Reserved

Message Delivery (server → client)

```
MESSAGE <message>
```

End of Messages Notification (server → client)

```
EOM
```

The `<message>` is a CSV encoded list of message components. The structure is identical to the message header received with the subscription acknowledgement.

5.2.8 Summary Request

Connected clients may request summary information from the server.

Request a Summary (client → server)

```
SUMMARY <client id>
```

Request a Summary (server → client)

```
SUMMARY <summary>
```

The `<summary>` is a CSV encoded list of summary components. Details TBD

5.2.9 Client/Client Communication

Connected clients can forward messages to other clients.

Forward a message (client → client)

```
FORWARD <client_pattern>@<host_pattern> <string>
```

Forward a string to clients at hosts specified by regular expressions.

Receive a forwarded message.

```
FORWARDED <client id> <string>
```

Receive a forwarded string from client identified by its id.

5.2.10 Session Notification

Connected clients can notify the logger server's archive about sessions.

Notify "create session" (client -> server)

CREATE_SESSION <string>

Inform server's archive that a session named by string was created.

Notify "close session" (client -> server)

CLOSE_SESSION <string>

Notify server's archive about closed session.

Copyright by EADS - All Rights Reserved